

ПРОЕКТИРОВАНИЕ МОДУЛЯ ХРАНЕНИЯ ДАННЫХ ДЛЯ ГИБРИДНОЙ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПОКАЗАТЕЛЕЙ СОЦИАЛЬНО-ЭКОНОМИЧЕСКОГО РАЗВИТИЯ РФ

УДК 004.9

Ольга Викторовна Китова,
д.э.н., профессор, зав. каф. Информатики
Российского экономического университета имени Г.В. Плеханова (РЭУ им. Г.В.
Плеханова)

Тел.: (499) 237-85-20
Эл. почта: olga.kitova@mail.ru

Ярослава Валерьевна Кривошеева,
аспирант каф. Информатики Российского
экономического университета имени Г.В.
Плеханова (РЭУ им. Г.В. Плеханова)
Тел.: (495) 794-00-96
Эл. почта: yaroslava2009@mail.ru

В статье описывается единый модуль хранения данных для гибридной системы прогнозирования показателей социально-экономического развития РФ. Излагается один из основных подходов к проектированию моделей данных – метод «нисходящего» проектирования. Рассматриваются примеры применения современных информационных технологий (ORM и MVC) к построению гибридной системы.

Ключевые слова: проектирование баз данных, метод «нисходящего» проектирования, гибридная система, ORM, MVC.

Olga V. Kitova
PhD in Economics, Professor, the head of
the chair of Informatics, Plekhanov Russian
University of Economics (PRUE)
Tel.: (499) 237-85-20
E-mail: olga.kitova@mail.ru

Yaroslava V. Krivosheeva
Post-graduate of the chair of Informatics,
Plekhanov Russian University of Economics
(PRUE)
Tel.: (495) 794-00-96
E-mail: yaroslava2009@mail.ru

DESIGNING DATABASE MODULE OF A HYBRID SYSTEM FOR FORECASTING OF INDICATORS OF SOCIO-ECONOMIC DEVELOPMENT OF THE RUSSIAN FEDERATION

A unified database module of a hybrid system for forecasting of indicators of socio-economic development of the Russian Federation is described in the article. One of the base approaches of designing databases (“top-down” method) is explained. Examples of applying modern information technologies (ORM and MVC) are considered.

Keywords: designing databases, “top-down” method of designing databases, hybrid system, ORM, MVC.

1. Введение

На кафедре информатики РЭУ им. Г.В. Плеханова разработан программно-технологический комплекс (далее – ПТК) для решения систем взаимосвязанных регрессионных уравнений, реализованный на платформе MS Excel с применением возможностей встроенного в платформу VBA. [5]

Регрессионные уравнения строятся на основе переменных, являющихся выбранными макроэкономическими показателями и описывающих фазовое состояние экономической системы. Таким образом, посредством регрессионных уравнений происходит связывание переменных между собой и формальное их разделение на зависимые и независимые (объясняющие). Далее происходит программное решение системы построенных регрессионных уравнений: нахождение коэффициентов при независимых переменных по методу наименьших квадратов, а также критерии точности и качества прогноза зависимых переменных. Процесс улучшения получаемого прогноза осуществляется итерационно: по результатам анализа полученных характеристик системы регрессионных уравнений, а также на основе анализа исходных данных принимается решение о соответствующих изменениях в регрессионных уравнениях и процесс решения системы регрессионных уравнений повторяется. [3]

В настоящее время в ПТК включается блок решения системы нейросетевых уравнений, построенных по аналогии с регрессионными уравнениями. В систему включаются те уравнения, по которым не удалось добиться удовлетворительных результатов методами регрессионного анализа. [4]

Для блока решения регрессионных уравнений все данные хранятся с помощью средств MS Excel. Однако в связи с тем, что система развивается и дополняется новыми блоками, возникла необходимость разработки единой глубоко продуманной структуры хранения данных.

В данной статье описана принципиально новая модель хранения данных, использующаяся в работе создаваемой гибридной системы.

Актуальность статьи заключается в необходимости разработки единой структуры хранения данных, которая будет использоваться как основа для осуществления прогноза как методами регрессионного анализа, так и с помощью технологий нейронных сетей. Это позволит исключить ошибки, обусловленные человеческим фактором (например, неправильный ввод данных, нарушающий общую логику работы системы), а также общей запутанностью существующего хранения данных, отсутствием средств поддержания целостности данных. [1]

2. Проектирование модуля хранения данных для гибридной системы

В теории баз данных существует ряд методов разработки моделей БД, отображающих разные уровни её архитектуры. Распространены два основных подхода к проектированию баз данных: «нисходящий» и «восходящий». [2]

Для модели данных гибридной системы был выбран метод нисходящего проектирования. При «нисходящем» проектировании осуществляется структурное проектирование сверху-вниз. И первым шагом здесь является описание предметной области посредством выделения значимых существенных, их атрибутов и взаимосвязей. На основе описания внешнего уровня строится концептуальная информационно-логическая модель предметной области (ИЛМ), затем на её основе получают даталогическую модель

(ДЛМ) базы данных. В даталогической сфере рассматриваются вопросы представления предметной области (описанной в информационной сфере) с помощью структур данных, определяемых выбором СУБД. В настоящее время наиболее широко для формирования даталогической сферы используются реляционные СУБД. ДЛМ является основой для следующего этапа проектирования БД – этапа формирования физической модели базы данных. [2]

Проанализировав существующую организацию хранения данных, использующуюся в ПТК, можно выделить следующие сущности и их атрибуты:

1. Показатели
 - Уникальный идентификатор показателя
 - Наименование показателя
 - Описание показателя
 - Единица измерения показателя
 - Признак показателя: сценарный/не сценарный
 - Идентификатор блока
2. Сфера, к которой относится показатель
 - Уникальный идентификатор сферы
 - Наименование сферы показателей
 - Описание сферы показателей
3. Блок, к которому относится показатель
 - Наименование блока
 - Описание блока
 - Идентификатор сферы

4. Уравнения

- Уникальный идентификатор уравнения
- Признак использования уравнения в дальнейших расчётах
- Шаг периода, по которому будут вычисляться прогнозные значения показателя (год, квартал, месяц)
- Идентификатор показателя

5. Аргументы уравнений

- Уникальный идентификатор аргумента
- Идентификатор уравнения
- Идентификатор показателя
- Признак использования аргумента в уравнении
- Величина лага аргумента

6. Фактические данные

- Уникальный идентификатор фактического значения
- Дата и время сделанного наблюдения
- Значение сделанного наблюдения
- Идентификатор показателя

7. Прогнозные данные

- Уникальный идентификатор прогнозного значения
- Версия прогноза
- Дата и время наблюдения
- Тип прогноза (регрессия/нейронная сеть)
- Значение полученного прогноза с помощью регрессии
- Значение полученного прогноза с помощью нейронных сетей
- Относительная ошибка прогноза, полученного с помо-

щью регрессии

- Относительная ошибка прогноза, полученного с помощью нейронных сетей
- Идентификатор уравнения

На рис. 1 представлена описанная концептуальная схема.

На рисунке 1 сущности показаны с помощью овального блока, атрибуты сущностей перечислены ниже самих сущностей. Связь «1-М» означает «один-ко-многим». Например, каждому показателю может соответствовать много уравнений, но каждому уравнению соответствует только один показатель.

Логическая модель описанной ранее концептуальной схемы данных представлена на рисунке 2.

Логическая модель данных уже привязывается к конкретной СУБД. В данном случае для удобства визуализации был использован MS Access.

Сущности концептуальной схемы превратились в реляционные таблицы, атрибуты – в поля таблиц. Связи «1-М» стали соответствовать реляционным связям «один-ко-многим».

Перечислим таблицы, представленные на рис. 2:

1. spr_Variable – сущность «Показатель»
2. spr_Block – сущность «Блок»
3. spr_Sphere – сущность «Сфера»
4. tbl_Arguments – сущность «Аргументы»
5. tbl_Equation – сущность «Уравнения»
6. tbl_Data_Value_fact – сущность «Фактические данные»

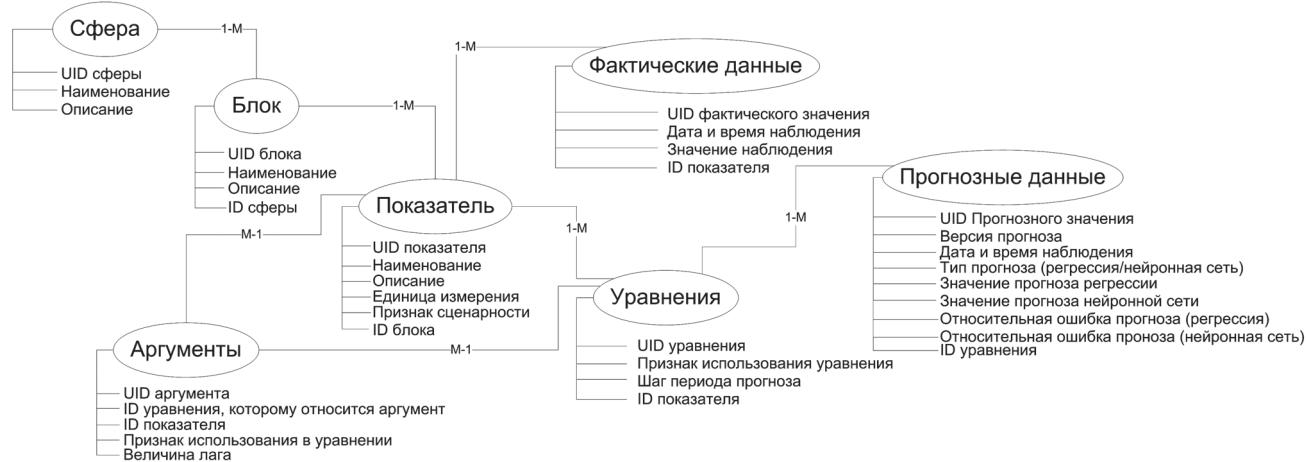


Рис. 1. Концептуальная схема модели данных

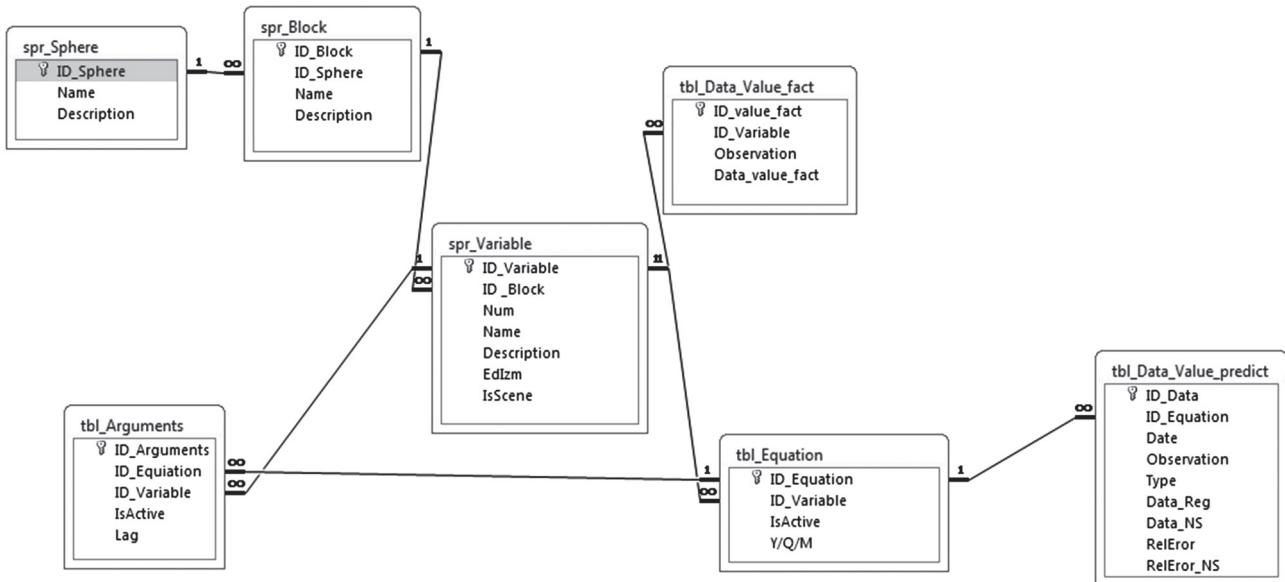


Рис. 2. Логическая модель данных

7. `tbl_Data_Value_predict` – сущность «Прогнозные данные»

Поля таблиц полностью соответствуют описанным ранее атрибутам.

Физическая модель данных предполагает конкретную реализацию на выбранной СУБД всех выделенных реляционных отношений. Рассмотрим пример такой реализации. В качестве примера возьмем основную сущность – «Показатель».

На СУБД PostgreSQL физическая реализация таблицы `spr_Variable` выглядит следующим образом [8]:

```
CREATE TABLE interface_spr_
variables
(
  «ID_variable» serial NOT NULL,
  «Num_excel» integer,
  «Name» character varying(200)
  NOT NULL,
  «Description» character
  varying(200),
  «EdIzm» character varying(20),
  «IsScene» boolean NOT NULL,
  «ID_block_id» integer NOT
  NULL,
  CONSTRAINT interface_spr_
variables_pkey PRIMARY KEY
  («ID_variable»),
  CONSTRAINT «interface_ID_
block_id_4df5acdf_fk_interface_
spr_block_ID_block» FOREIGN
  KEY («ID_block_id»)
    REFERENCES interface_spr_
  block («ID_block») MATCH
    SIMPLE
```

ON UPDATE NO ACTION
ON DELETE NO ACTION
DEFERRABLE INITIALLY
DEFERRED,

CONSTRAINT «interface_spr_variables_Name_key» UNIQUE («Name»)

1

```
CREATE INDEX «interface_spr_variables_Name_4792f840_like»
  ON interface_spr_variables
  USING btree
  («Name» COLLATE pg_catalog.»default» varchar_
  pattern_ops);
```

В данном случае создается таблица с конкретным указанием ее полей и типов полей. Также к таблице создается индекс. Связь «один-ко-многим» реализуется с помощью внешнего ключа (FOREIGN KEY).

Для других существ были созданы аналогичные таблицы.

3. Применение технологии ORM

Дополнительный блок решения систем нейросетевых уравнений гибридной системы в настоящее время реализуется с помощью фреймворка Django на языке программирования Python. Фреймворк использует технологию ORM (object-relational mapping или объектно-реляционное отображение). Данная технология связывает базы данных с концепциями объектно-ориентирован-

ных языков программирования, создавая «виртуальную объектную базу данных». ORM избавляет программиста от написания большого количества кода, часто однообразного и подверженного ошибкам, тем самым значительно повышая скорость разработки. Кроме того, большинство современных реализаций ORM позволяют программисту при необходимости самому жёстко задать код SQL-запросов, который будет использоваться при тех или иных действиях (сохранение в базу данных, загрузка, поиск и т. д.) с постоянным объектом. [6, 7]

ORM удобно использовать совместно с технологией MVT (или MVC). Model-view-controller («модель-представление-контроллер») – схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Основная цель применения этой концепции состоит в разделении бизнес-логики (модели) от её визуализации (представления, вида). За счет такого разделения повышается возможность повторного использования. Наиболее полезно применение данной концепции в

тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения. В частности, выполняются следующие задачи:

- К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели. Например, некоторые данные могут быть одновременно представлены в виде электронной таблицы, гистограммы и круговой диаграммы.
- Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных), для этого достаточно использовать другой контроллер.

• Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (модели), вообще не будут осведомлены о том, какое представление будет использоваться.

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- Модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.

- Представление, вид (англ. View). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.

- Контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции. [6]

Рассмотрим пример применения технологии ORM на спроектированной модели данных.

Переложение части реляционных таблиц в объекты на языке Python в контексте Django будет выглядеть следующим образом:

```
class spr_Variables(models.Model):
    ID_variable = models.AutoField(primary_key=True)
    ID_block = models.ForeignKey(spr_Block)
    Num_excel = models.IntegerField(null=True)
    Name = models.CharField(max_length=200,unique=True)
    Description = models.CharField(max_length=200,null=True)
    EdIzm = models.CharField(max_length=20,null=True)
    IsScene = models.BooleanField(default=False)

    def __unicode__(self):
        return self.Name
```

```
class tbl_Value_fact(models.Model):
    ID_value_f = models.AutoField(primary_key=True)
    ID_variable = models.ForeignKey(spr_Variables)
    Observation = models.DateTimeField()
    Value_fact = models.FloatField(null=True)

    def __unicode__(self):
        return str(self.Value_fact)
```

```
class tbl_Equation(models.Model):
    ID_equation = models.AutoField(primary_key=True)
    ID_variable = models.ForeignKey(spr_Variables)
    IsEqActive = models.BooleanField(default=True)
    Y_Q_M_choices = models.CharField(max_length=1, choices=(('Y','Year'),('Q','Quarter'),('M','Month')),default='Q')

    def __unicode__(self):
        return str(self.ID_variable)
```

Классами удобно оперировать при выполнении различных программных расчетов. Описанные классы соответствуют сущностям концептуальной схемы, реляционным отношениям логической схемы данных и созданным таблицам физической модели данных.

4. Заключение

Разработка и поддержание жизнеспособности большого программно-технологического комплекса – задача нетривиальная. В связи с этим в процессе прогноза, который

реализует ПТК, постоянно возникают места, требующие усовершенствования. В данной статье описано такого рода усовершенствование, связанное с проектированием единой структуры хранения данных.

Результаты этой работы будут использованы не только для усовершенствования существующего ПТК, но и в реализуемом в настоящий момент дополнении ПТК – блоке решения нейросетевой системы уравнений.

Однако следует отметить, что предложенная модель данных не является окончательной. Планируется в дальнейшем ее совершенствовать и расширять с целью хранения большего количества данных для осуществления качественного анализа получаемых прогнозов экономических показателей.

Литература

1. Интернет-технологии в экономике знаний: Учебник / под науч. ред. Н.М. Абдиева – М.: ИНФРА-М, 2010. – 448 с.
2. Голицына О.Л., Максимов Н.В., Попов И.И. Базы Данных. Учебное пособие. – 2-е изд., испр. и доп. – М.: Форум, Инфра-М, 2009. – 400 с.: ил.
3. Китова О.В., Дьяконова Л.П., Колмаков И.Б., Кривошеева Я.В. Разработка и применение программно-технологических средств для автоматизированной отладки показателей качества регрессионных уравнений // Журнал «Менеджмент и бизнес-администрирование». – № 3. – 2015. – С. 107-111.
4. Шарафутдинова А. Р. Применение нейросетевых моделей для повышения качества и точности краткосрочного прогноза показателей социально-экономического развития России // Вестник Российской экономического университета им. Г.В. Плеханова. – 2012. – № 08. – С. 102-109.
5. Свидетельство о государственной регистрации базы данных «База данных регрессионных уравнений для прогнозирования показателей инвестиций экономики РФ» № 2015620810 Зарегистрировано в Реестре баз данных 26.05.2015 / Авторы - правообладатели Колмаков И.Б., Потапов С.В., Пеньков И.А.

6. <http://djbook.ru/> - сайт документации Django.
7. <http://python.org/> – официальный сайт Python.
8. <http://www.postgresql.org/> – Документация PostgreSQL.

References

1. Internet technology in the knowledge economy: schoolbook / edited by N.M. Abdikeev – M.: INFRA-M, 2010. – 448 pp.
2. Golizyna O.L., Maksimov N.V., Popov I.I. Databases. Ucheb. posobie. – 2-e izd., ispr. and dop. – M.: Forum, INFRA-M, 2009. – 400 pp.: ill.
3. Kitova O.V., D'jakonova L.P., Kolmakov I.B., Krivosheeva Y.V. Development and application of soft-ware technology tools for automated debugging of quality indicators of the regression equations // Magazine “Management and business-administration”, № 3/2015, pp. 107-111.
4. Sharafutdinova A.R. Applying of neural models for raising of quality and accuracy of short-term forecast of indicators of socio-economic development of the Russian Federation // Vestnik of the Plekhanov Russian University of Economics, № 8/2012, pp. 102–109.
5. Certificate of the state registration of databases No. 2015620810 «Regression equations database for the Russian economy investment indicators predicting». Registered in the register of databases 26.05.2015 / Rightholders Kolmakov I.B., Potapov S.V., Penkov I.A.
6. <http://djbook.ru/> – the official website of Django documentation.
7. <http://python.org/> – the official website of Python.
8. <http://www.postgresql.org/> – the official website of PostgreSQL documentation.